Versioning in the Trigger Database

a component of keeping track of the online code

Elizabeth Gallas

Fermilab

D0 Rug Meeting

October 25, 2000

Trigger Database

- L1L2
  - People: Elizabeth G, Joe Kuah
  - L1L2_TRIGGER (DB)
    - on development platform
    - contains one trigger list for cal sim and additional global AO terms
  - display interface keeping pace with db changes (MISWEB application)
  - entry interface using db server
    - have L3 application running (MI) - understand this and explore other options.
  - Display interface needed for many views of the data
- L3
  - People: Amber B, Barb A (OD), Carmem S(interface) till Nov 1
  - TRIGGER_DB
    - on development platform
    - some changes must be retrofitted back into Oracle Designer
    - contains some data currently in use
  - entry interface (dbserver, JAVA)
    - PC based package included
      - ability to define standard refsets
      - ability to copy/modify these refsets to form a 'L3 trigger list'.

Both database/interfaces are in development (see next slide)

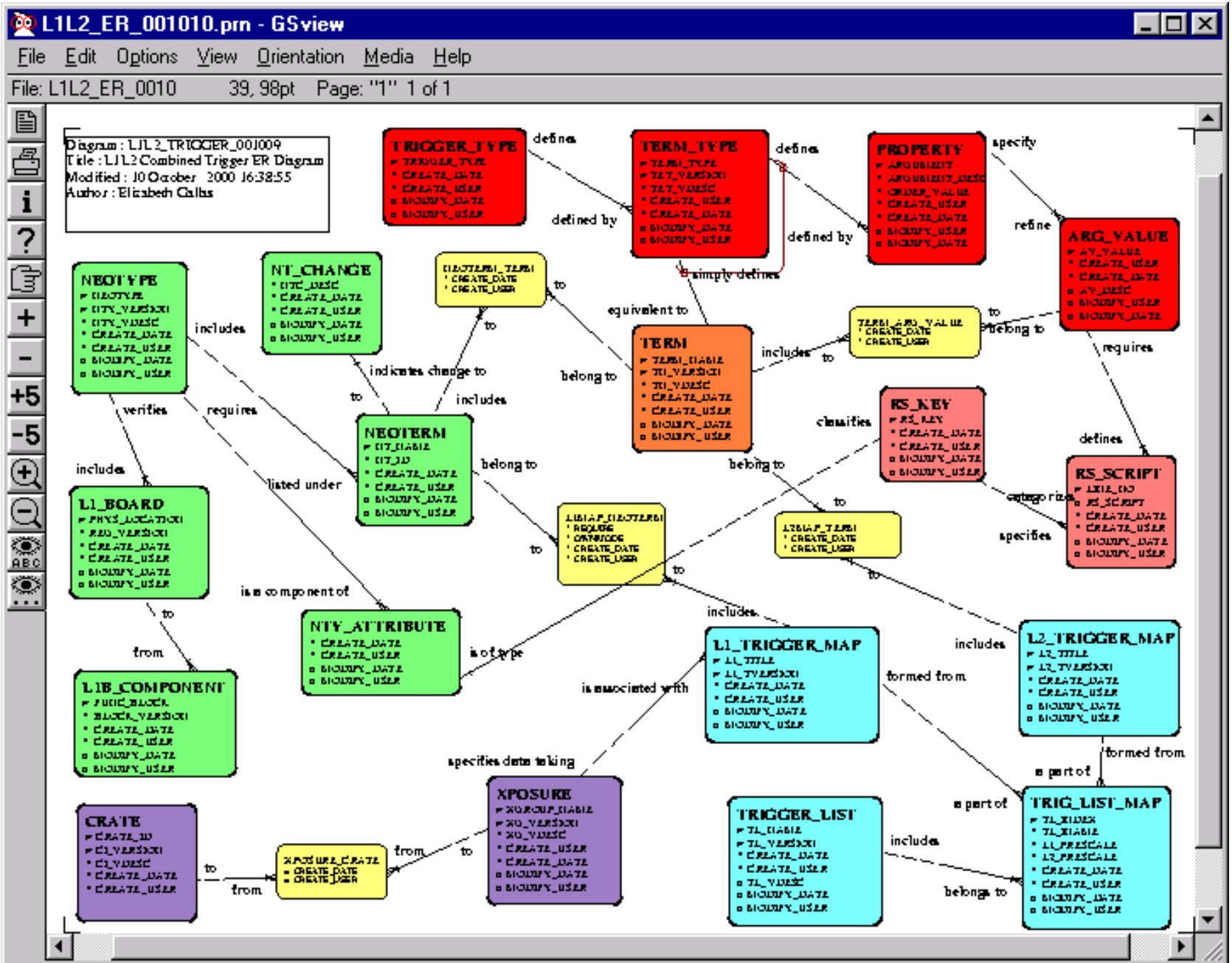
Current Projects/Issues

- Integration with L3 trigger database
- Versioning at L1
- Other L1L2 database design issues that must be resolved
  - L1 cal, L1 pseudo terms
  - L2 versioning
  - Shared access to Hardware Database
  - Exposure groups/crates and how they relate to the Luminosity Database, Run Summary Database
  - online/offline transfer of trigger lists
- Interfaces in development:
  - pass a trigger list to COOR
  - store and retrieve trigger information using the L3 interface as a prototype
  - display trigger lists and many other views

L1L2 changes (versioning anything that can affect the trigger decision)

- L1L2 versions will be used by
  - COOR to check registers/processors of online components with versions in database (CTT used as prototype - up to 100 registers checked)
  - online examine - check firmware versions written to the data-stream
- Reasons for L1L2 version changes
  - Physics, Algorithm improvements, Bug fixes, Firm/soft-ware changes
  - Firmware change (L1 and L1-like components input to L2)
    - database includes pointers to which NEOTERMS are affected.
  - Other versions
    - currently capable of L2 global object versioning but more versions in L2 administrator/workers (preprocessors) may need to be checked
    - L1 framework or other trigger element 'versions' to store as well (anything that might affect trigger decision outcome)

L1L2 ER Diagram



Database Versions vs Repository Versions and how they relates to real components

- All agree: Source code should be stored in CVS repository
  - Accurate bookkeeping is required at each level
    - Entries of versions in the trigger database need to be considered carefully
    - CVS-like tag numbers and a written comment will be stored in trigger DB
    - The relevant CVS package should contain only the code used online
    - Packages must be organized in a way such that others can understand how it is distributed to the online components (ie comes with documentation)
    - side note: source can be put into repository from NT
- Binaries
  - All agree: CVS not ideal (differences are stored, not complete copies)
  - If these are short term backups, store someplace else
  - CVS storage - useless long term because other things change
  - If the source/data is stored, why can't binaries be regenerated?

“Trigger Certification” - a process required for all changes in the online trigger

- Idea to change trigger
- Test/time algorithms in simulation
- Full evaluation of
  - which trigger terms are effected,
  - evaluation of efficiencies, etc.,
  - how will changes be cross checked online
- Approval of change (trigger certification board)
- Put new trigger list into the Trigger Database
- New trigger list is used online

Effect (Affect) on Simulation

- Jerry suggests
  - every piece of firmware has 2 tags
    - online register
    - simulation version corresponding to that trigger version
- Looking for advice on
  - dynamic load libraries
    - (may only work if there are no interface changes)
  - multiple executables (releases?)
    - is there something we can do now to reduce the number of exe's?